warp between two images of a sequence. Such a technique can be extended to 3D (See, F. F. Pighin, J. Hecker, D. Lischinski, R. Szeliski, and D. H. Salesin. Synthesizing Realistic Facial Expressions from Photographs. In SIGGRAPH 98 Conference Proceedings, pages 75–84. July 1998.).

6. Other approaches such as control points and finite element models.

For these techniques, facial sensing enhances the animation process by providing automatic extraction and characterization of facial features. Extracted features may be used to interpolate expressions in the case of key framing and interpolation models, or to select parameters for direct parameterized models or pseudo-muscles or muscles models. In the case of 2-D and 3-D morphing, facial sensing may be used to automatically select features on a face providing the appropriate information to perform the geometric transformation.

An example of an avatar animation that uses facial feature tracking and classification may be shown with respect to FIG. **15**. During the training phase, the individual is prompted for a series of predetermined facial expressions (block **120**), and sensing is used to track the features (block **122**). At predetermined locations, jets and image patches are extracted for the various expressions (block **124**). Image patches surrounding facial features are collected along with the jets **126** extracted from these features. These jets are used later to classify or tag facial features **128**. This is done by using these jets to generate a personalized bunch graph and by applying the classification method described above.

As shown in FIG. **16**, for animation of an avatar, the system transmits all image patches **128**, as well as the image of the whole face **130** (the "face frames") minus the parts shown in the image patches to a remote site (blocks **132** & **134**). The software for the animation engine also may need to be transmitted. The sensing system then observes the user's face and facial sensing is applied to determine which of the image patches is most similar to the current facial expression (blocks **136** & **138**). The image tags are transmitted to the remote site (block **140**) allowing the animation engine to assemble the face **142** using the correct image patches.

To fit the image patches smoothly into the image frame, Gaussian blurring may be employed. For realistic rendering, local image morphing may be needed because the animation may not be continuous in the sense that a succession of images may be presented as imposed by the sensing. The morphing may be realized using linear interpolation of corresponding points on the image space. To create intermediate images, linear interpolation is applied using the following equations:

$$P_i=(2-i)P_1+(i-1)P_2 \quad (7)$$

$$I_i=(2-i)I_1+(i-1)I_2 \quad (8)$$

where P_1 and P_2 are corresponding points in the images I_1 and I_2, and I_i is the i^{th} interpolated image: with $1 \leqq i \leqq 2$. Note that for process efficient, the image interpolation may be implemented using a pre-computed hash table for P_i and I_i. Based on the number and accuracy of points used, and their accuracy, the interpolated facial model generally determines the resulting image quality.

Thus, the reconstructed face in the remote display may be composed by assembling pieces of images corresponding to the detected expressions in the learning step. Accordingly, the avatar exhibits features corresponding to the person commanding the animation. Thus, at initialization, a set of cropped images corresponding to each tracked facial feature and a "face container" as the resulting image of the face after each feature is removed. The animation is started and facial sensing is used to generate specific tags which are transmitted as described previously. Decoding occurs by selecting image pieces associated with the transmitted tag, e.g., the image of the mouth labeled with a tag "smiling-mouth" **146** (FIG. **16**).

A more advanced level of avatar animation may be reached when the aforementioned dynamic texture generation is integrated with more conventional techniques of volume morphing as shown in FIG. **17**). To achieve volume morphing, the location of the node positions may be used to drive control points on a mesh **150**. Next, the textures **152** dynamically generated using tags are then mapped onto the mesh to generate a realistic head image **154**. An alternative to using the sensed node positions as drivers of control points on the mesh is to use the tags to select local morph targets. A morph target is a local mesh configuration that has been determined for the different facial expressions and gestures for which sample jets have been collected. These local mesh geometries can be determined by stereo vision techniques. The use of morph targets is further developed in the following references community (see, J. R. Kent, W. E. Carlson, and R. E. Parent, Shape Transformation for Polyhedral Objects, In SIGGRAPH 92 Conference Proceedings, volume 26, pages 47–54, August 1992; Pighin et al. 1998, supra).

A useful extension to the vision-based avatar animation is to integrate the facial sensing with speech analysis in order to synthesize the correct lip motion as shown in FIG. **18**. The lip synching technique is particularly useful to map lip motions resulting from speech onto an avatar. It is also helpful as a back-up in case the vision-based lip tracking fails.

Although the foregoing discloses the preferred embodiments of the present invention, it is understood that those skilled in the art may make various changes to the preferred embodiments without departing from the scope of the invention. The invention is defined only the following claims.

What is claimed is:

1. A method for feature sensing on a sequence of image frames, comprising:

a step for transforming each image frame using a wavelet transformation to generate a transformed image frame;

a step for initializing nodes of a model graph, each node associated with a wavelet jet specific to a feature, to locations on the transformed image frame by moving the model graph across the transformed image frame and placing the model graph at a location in the transformed image frame of maximum jet similarity between the wavelet jets of the nodes and locations on the transformed image frame determined as the model graph is moved across the transformed image frame;

a step for tracking the location of one or more nodes of the model graph between image frames; and

a step for reinitializing the location of a tracked node if the tracked node's location deviates beyond a predetermined position constraint between image frames.

2. A method for feature sensing as defined in claim **1**, wherein the model graph used in the initializing step is based on a predetermined pose.

3. A method for feature sensing as defined in claim **1**, wherein the tracking step tracks the node locations using elastic bunch graph matching.

4. A method for feature sensing as defined in claim **1**, wherein the tracking step uses linear position prediction to